UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/894,260	06/28/2001	Richard James Eickemeyer	ROC 9 2000 0282 US1	5055

7590 01/12/2005

Robert R. Williams
IBM Corporation, Dept. 917
3605 Highway 52 North
Rochester, MN 55901-7829

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 01/12/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/894,260	EICKEMEYER ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 July 2004 and 25 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-13 and 15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-13 and 15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 3-13, and 15 have been considered. Claims 2 and 14 have been cancelled as per Applicant's request. Claims 1, 5-7, 9-11, 13, and 15 have been amended as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as filed 29 July 2004 and Amendment as filed 25 October 2004.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 3-4 and 7-9 are rejected under 35 U.S.C. 102(e) as being taught by Koblenz et al., U.S. Patent Number 6,353,829 (herein referred to as Koblenz).

5. Referring to claim 1, Koblenz has taught a resource queue, comprising:

   a. A plurality of entries, each entry having unique resources required for information processing (Koblenz column 6, lines 15-34 and column 7, line 56 to column 8, line 15);

   b. The plurality of entries allocated amongst a plurality of independent simultaneously executing hardware threads such that resources of more than one

thread may be within the queue (Koblenz column 5, lines 23-37; column 6, lines 15-34; and column 7, line 56 to column 8, line 15); and

c. A portion of the plurality of entries being allocated to one thread and being capable of being interspersed among another portion of the plurality of entries allocated to another thread (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14). In regards to Koblenz, the entries are independently allocated without regards to what threads the other entries belong to, the system just searches for an open entry of appropriate size, thereby interspersing the data.

d. Wherein a first entry of one thread is capable of wrapping around a last entry of the same thread to access an available entry (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14).

6. Referring to claim 3, Koblenz has taught

a. A head pointer and a tail pointer for at least one thread wherein the head pointer is the first entry of the at least one thread and the tail pointer-is the last entry of the at least one thread (Koblenz column 11, lines 10-55 and Figure 4), and

b. One of the unique resources is a bank number to indicate how many times the head pointer has wrapped around the tail pointer in order to maintain an order of the resources for the at least one thread (Koblenz column 11, lines 10-55; column

14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 4; Figure 11; and Figure 14).

7. Referring to claim 4, Koblenz has taught at least one free pointer for the at least one thread indicating an entry in the queue available for resources of the at least one thread (Koblenz column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14).

8. Referring to claim 7, Koblenz has taught a method of allocating a shared resource queue for simultaneous multithreaded electronic data processing, comprising:

a. Determining if the shared resource queue is empty for a particular thread (Koblenz column 6, lines 15-34; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14);

b. Finding a first entry of a said particular thread (Koblenz column 6, lines 15-34; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14);

c. Determining if the first entry and a free entry of the particular thread are the same(Koblenz column 6, lines 15-34; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14);

d. If, not advancing the first entry to the free entry (Koblenz column 6, lines 15-34; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14);

e. Incrementing a bank number if the first entry passes a last entry of the particular thread before it finds the free entry (Koblenz column 6, lines 15-34; column 14,

line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14);

f. Allocating the next free entry by storing resources for the particular thread (Koblenz column 6, lines 15-34; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14).

9. Referring to claim 8, Koblenz has taught

a. Locating the last entry in the shared resource queue pertaining to the particular thread (Koblenz column 6, lines 15-34; column 13, line 11 to column 14, line 17; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figures 9A-9E; Figure 11; and Figure 14);

b. Determining if the last entry is also the first entry for the particular thread (Koblenz column 6, lines 15-34; column 13, line 11 to column 14, line 17; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figures 9A-9E; Figure 11; and Figure 14);

c. If not, finding the next entry pertaining to the particular thread (Koblenz column 6, lines 15-34; column 13, line 11 to column 14, line 17; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figures 9A-9E; Figure 11; and Figure 14);

d. Determining if the bank number of the next entry is the same as the last entry and if so, deallocating the next entry by marking the resources as invalid (Koblenz column 6, lines 15-34; column 13, line 11 to column 14, line 17; column 14, line

54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figures 9A-9E; Figure 11; and Figure 14); and

e. If not, then skipping over the next entry and decrementing the bank number (Koblenz column 6, lines 15-34; column 13, line 11 to column 14, line 17; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figures 9A-9E; Figure 11; and Figure 14);

f. Finding the next previous entry pertaining to the particular thread (Koblenz column 6, lines 15-34; column 13, line 11 to column 14, line 17; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figures 9A-9E; Figure 11; and Figure 14).

10. Referring to claim 9, Koblenz has taught

a. Setting a flush point indicative of an oldest entry to be deallocated pertaining to the particular thread (Koblenz column 6, lines 35-47; column 13, line 11 to column 14, line 17; and Figures 9A-9E); and

b. Invalidating all entries between a head pointer and the flush point which have the same and greater bank number than the bank number of the flush point (Koblenz column 6, lines 35-47; column 13, line 11 to column 14, line 17; and Figures 9A-9E).

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person

> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Koblenz et al., U.S. Patent Number 6,353,829 (herein referred to as Koblenz), Koblenz et al., U.S. Patent Number 6,353,829 (herein referred to as Koblenz), as applied to claim 1 above, in view of Lee et al., U.S. Patent Number 6;629,271 (herein referred to as Lee). Koblenz has not taught an out-of-order computer processor and the resource queue may further comprise a load reorder queue and/or a store reorder queue and/or a global completion table and/or a branch information queue. Lee has taught an out-of-order computer processor (Lee column 5, lines 35-37 and Figure 1, element 100) and the resource queue may further comprise a load reorder queue and/or a store reorder queue and/or a global completion table and/or a branch information queue (Lee column 2, line 49; column 4, lines 66-67; column 5, lines 3-7 and 33-41; and Figure 2). A person of ordinary skill in the art would have recognized at the time the invention was made that out-of-order execution reduces idle cycles within a thread, thereby increasing the speed and efficiency of the device. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the out-of-order methods of Lee in the device of Koblenz to improve processor speed and efficiency.

13. Claim 6, 10-13, and 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lee et al., U.S. Patent Number 6,629,271 (herein referred to as Lee) in view of Koblenz et al., U.S. Patent Number 6,353,829 (herein referred to as Koblenz), Koblenz et al., U.S. Patent Number 6,353,829 (herein referred to as Koblenz).

14. Referring to claim 6, Lee has taught an out-of-order multithreaded computer processor, comprising:

a. A load reorder queue (Lee column 2, line 49; column 4, lines 66-67; column 5, lines 3-7 and 33-41; and Figure 2);

b. A store reorder queue (Lee column 2, line 49; column 4, lines 66-67; column 5, lines 3-7 and 33-41; and Figure 2);

c. A global completion table (Lee column 2, line 49; column 4, lines 66-67; column 5, lines 3-7 and 33-41; and Figure 2);

d. A branch information queue (Lee column 2, line 49; column 4, lines 66-67; column 5, lines 3-7 and 33-41; and Figure 2),

15. Lee has not taught at least one of the queues being a resource queue comprising:

a. A plurality of entries, each entry having unique resources required for information processing;

b. The plurality of entries allocated amongst a plurality of independent simultaneously executing hardware threads such that resources of more than one thread may be within the queue; and

c. A portion of the plurality of entries being allocated to one thread and being capable of being interspersed among another portion of the plurality of entries allocated to another thread;

d. A first entry of one thread being capable of wrapping around a last entry of the same thread;

e. A head pointer and a tail pointer for at least one thread wherein the head pointer is the first entry of the at least one thread and the tail pointer is the last entry of the at least one thread;

f. A bank number to indicate how many times the head pointer has wrapped around the tail pointer in order to maintain an order of the resources for the at least one thread; and

g. At least one free pointer for the at least one thread indicating an entry in the queue available for resources of the at least one thread.

16. Koblenz has taught at least one of the queues being a resource queue comprising:

a. A plurality of entries, each entry having unique resources required for information processing (Koblenz column 6, lines 15-34 and column 7, line 56 to column 8, line 15);

b. The plurality of entries allocated amongst a plurality of independent simultaneously executing hardware threads such that resources of more than one thread may be within the queue (Koblenz column 5, lines 23-37; column 6, lines 15-34; and column 7, line 56 to column 8, line 15); and

c. A portion of the plurality of entries being allocated to one thread and being capable of being interspersed among another portion of the plurality of entries allocated to another thread (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14). In regards to Koblenz, the entries are independently allocated without regards to what threads the other entries belong to, the system just searches for an open entry of appropriate size, thereby interspersing the data.

d. Wherein a first entry of one thread is capable of wrapping around a last entry of the same thread to access an available entry (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14).

e. A head pointer and a tail pointer for at least one thread wherein the head pointer is the first entry of the at least one thread and the tail pointer is the last entry of the at least one thread (Koblenz column 11, lines 10-55 and Figure 4), and

f. One of the unique resources is a bank number to indicate how many times the head pointer has wrapped around the tail pointer in order to maintain an order of the resources for the at least one thread (Koblenz column 11, lines 10-55; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 4; Figure 11; and Figure 14).

g. At least one free pointer for the at least one thread indicating an entry in the queue available for resources of the at least one thread (Koblenz column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14).

17. A person of ordinary skill in the art at the time the invention was made, and as recognized by Koblenz, that the circular queue of Koblenz maximizes concurrent execution of memory routines and optimizes allocation of memory blocks (Koblenz column 6, lines 9-12), thereby increasing processor efficiency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Koblenz in the device of Lee to improve processor efficiency.

18. Referring to claim 10, Lee has taught a shared resource mechanism in a hardware multithreaded pipeline processor, said pipeline processor simultaneously processing a plurality of threads, said shared resource mechanism comprising:

a. A dispatch stage of said pipeline processor (Lee column 3, lines 57-59 and Figure 1, element 114);

b. At least one shared resource queue connected to the dispatch stage (Lee Figure 1, element 152 and Figure 2, element 78);

c. Dispatch control logic connected to the dispatch stage and to the at least one shared resource queue (Lee column 3, lines 57-61; column 4, lines 66-67; column 5, lines 6-7; and Figure 1, elements 112 and 114); and

d. An issue queue of said pipeline processor connected to said dispatch stage and to the at least one shared resource queue (Lee column 3, lines 39-46 and 57-60 and Figure 1, elements 112, 114, and 152);

19. Lee has not taught

a. Wherein the at least one shared resource queue allocates and deallocates resources for at least two of said plurality of threads passing into said issue queue in response to the dispatch control logic;

b. The at least one shared resource queue further comprises a plurality of entries allocated to one thread and capable of being interspersed among another plurality of entries allocated to another of the plurality of threads;

c. Wherein a first entry of one thread is capable of wrapping around a last entry of the same thread to access an available entry for allocating resources of the one thread.

20. Koblenz has taught

a. Wherein the at least one shared resource queue allocates and deallocates resources for at least two of said plurality of threads passing into said issue queue in response to the dispatch control logic (Koblenz column 6, lines 15-47);

b. The at least one shared resource queue further comprises a plurality of entries allocated to one thread and capable of being interspersed among another plurality of entries allocated to another of the plurality of threads (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14). In regards to Koblenz, the entries are independently allocated without regards to what threads the other entries belong to, the system just searches for an open entry of appropriate size, thereby interspersing the data.

c. Wherein a first entry of one thread is capable of wrapping around a last entry of the same thread to access an available entry for allocating resources of the one thread (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14).

21. A person of ordinary skill in the art at the time the invention was made, and as recognized by Koblenz, that the circular queue of Koblenz maximizes concurrent execution of memory

routines and optimizes allocation of memory blocks (Koblenz column 6, lines 9-12), thereby increasing processor efficiency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Koblenz in the device of Lee to improve processor efficiency.

22. Referring to claims 11 and 12, Lee has taught an apparatus to enhance processor efficiency, comprising:

a. Means to fetch instructions from a plurality of threads into a hardware multithreaded pipeline processor (Lee column 5, lines 15-16);

b. Means to distinguish said instructions into one of a plurality of threads (Lee Figure 2, element 78);

c. Means to decode said instructions (Lee column 3, lines 40-42);

d. Means to dispatch said instructions (Lee column 3, lines 57-61);

e. Means to execute said instructions and said resources for the one of said at least two threads (Lee column 4, lines 20-24).

23. Lee has not taught

a. Means to allocate a plurality of entries in at least one shared resource between at least two of the plurality of threads simultaneously executing (Applicant's claim 11);

b. Means to allocate and intersperse entries in the at least one shared resource to one thread among entries allocated to other threads (Applicant's claim 11);

c. Means for a first entry of one thread to wrap around a last entry of the same thread (Applicant's claim 11);

d. Means to determine if said instructions have sufficient private resources and at least one shared resource queue for dispatching said instructions (Applicant's claim 11);

e. Means to deallocate said entries in said at least one shared resource when one of said at least two threads are dispatched (Applicant's claim 11); and

f. Means to flush the at least one shared resource of all of said entries pertaining to the one of said at least two threads (Applicant's claim 12).

24. Koblenz has taught

a. Means to allocate a plurality of entries in at least one shared resource between at least two of the plurality of threads simultaneously executing (Koblenz column 6, lines 15-34 and column 7, line 56 to column 8, line 15);

b. Means to allocate and intersperse entries in the at least one shared resource to one thread among entries allocated to other threads (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14). In regards to Koblenz, the entries are independently allocated without regards to what threads the other entries belong to; the system just searches for an open entry of appropriate size, thereby interspersing the data.

c. Means for a first entry of one thread to wrap around a last entry of the same thread (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14);

d. Means to determine if said instructions have sufficient private resources and at least one shared resource queue for dispatching said instructions (Koblenz column 6, lines 15-34; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14);

e. Means to deallocate said entries in said at least one shared resource when one of said at least two threads are dispatched (Koblenz column 6, lines 35-47; column 13, line 11 to column 14, line 17; and Figures 9A-9E); and

f. Means to flush the at least one shared resource of all of said entries pertaining to the one of said at least two threads (Koblenz column 6, lines 35-47; column 13, line 11 to column 14, line 17; and Figures 9A-9E).

25. A person of ordinary skill in the art at the time the invention was made, and as recognized by Koblenz, that the circular queue of Koblenz maximizes concurrent execution of memory routines and optimizes allocation of memory blocks (Koblenz column 6, lines 9-12), thereby increasing processor efficiency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Koblenz in the device of Lee to improve processor efficiency.

26. Referring to claim 13, Lee has taught a computer processing system, comprising:

a. A central processing unit (Lee Figure 1);

b. A semiconductor memory unit attached to said central processing unit (Lee Figure 1);

c. At least one memory drive capable of having removable memory (Lee Figure 1); and

d. A hardware multithreading pipelined processor within said central processing unit to simultaneously process at least two independent threads of execution, said pipelined processor comprising a fetch stage, a decode stage, and a dispatch stage (Lee column 3, lines 39-43 and 57-59; column 5, lines 15-16 and 40; Figure 1; and Figure 2).

27. Lee has not taught

a. A keyboard/pointing device controller attached to said central processing unit for attachment to a keyboard and/or a pointing device for a user to interact with said computer processing system and

b. A plurality of adapters connected to said central processing unit to connect to at least one input/output device for purposes of communicating with other computers, networks, peripheral devices, and display devices.

28. "Official Notice" is taken that it was well known and expected in the art at the time of the invention to have a keyboard/pointing controller for attachment to a keyboard and/or a pointing device for a user to interact with said computer processing system and a plurality of adapters to connect to input/output devices for purposes of communicating with other computers, networks, peripheral devices and display devices attached to the central processing unit. Therefore it would have been obvious to one of ordinary skill at the time of the invention to have modified the Lee et al. reference by adding a keyboard/pointing controller for attachment to a keyboard and/or a pointing device for a user to interact with said computer processing system and a plurality of adapters to connect to input/output devices for purposes of communicating with other computers, networks, peripheral devices and display devices.

29. In addition, Lee has not taught

a. At least one shared resource queue within said central processing unit, said shared resource queue having a plurality of entries pertaining to more than one thread in which entries pertaining to different threads are interspersed among each other (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14). In regards to Koblenz, the entries are independently allocated without regards to what threads the other entries belong to, the system just searches for an open entry of appropriate size, thereby interspersing the data.

b. A head pointer (Koblenz column 11, lines 10-55 and Figure 4) pertaining to an entry of one thread is capable of wrapping around (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14) a tail pointer (Koblenz column 11, lines 10-55 and Figure 4) pertaining to another entry of the same one thread to access an available entry and the number of times the head pointer wraps around the tail pointer is recorded (Koblenz column 11, lines 10-55; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 4; Figure 11; and Figure 14).

30. Koblenz has taught

a. At least one shared resource queue within said central processing unit, said shared resource queue having a plurality of entries pertaining to more than one thread in which entries pertaining to different threads are interspersed among each other

(Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14). In regards to Koblenz, the entries are independently allocated without regards to what threads the other entries belong to, the system just searches for an open entry of appropriate size, thereby interspersing the data.

b. A head pointer (Koblenz column 11, lines 10-55 and Figure 4) pertaining to an entry of one thread is capable of wrapping around (Koblenz column 6, lines 15-34; column 7, line 56 to column 8, line 15; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 11; and Figure 14) a tail pointer (Koblenz column 11, lines 10-55 and Figure 4) pertaining to another entry of the same one thread to access an available entry and the number of times the head pointer wraps around the tail pointer is recorded (Koblenz column 11, lines 10-55; column 14, line 54 to column 15, line 25; column 16, line 37 to column 17, line 6; Figure 4; Figure 11; and Figure 14).

31. A person of ordinary skill in the art at the time the invention was made, and as recognized by Koblenz, that the circular queue of Koblenz maximizes concurrent execution of memory routines and optimizes allocation of memory blocks (Koblenz column 6, lines 9-12), thereby increasing processor efficiency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular queue of Koblenz in the device of Lee to improve processor efficiency.

32. Referring to claim 15, Lee has taught wherein the hardware multithreaded pipelined processor in the central processing unit is an out of order processor (Lee column 5, lines 35-37 and Figure 1, element 100).

Response to Arguments

33. Applicant's arguments with respect to claims 1, 3-13, and 15 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

34. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

35. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

36. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:30am-5:00pm.

37. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

38. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
07 January 2005